

---

# RSE Quick Start Guide

---

by

Susan Gantner & Jon Paris

Partner400 & System *i* Developer

(Version RDi 2.01)





## RSE Quick Start Guide

This Quick Start Guide is intended to make it a little easier and faster for you to begin exploring RSE (Remote System Explorer) as a far more productive replacement for SEU and PDM. It is by no means a complete look at the toolset and is meant to supplement rather than replace education, product documentation and further individual exploration.

We use the terms RSE and RDi interchangeably. RSE has been a significant portion of products with different names over the years, such as RDP and WDSC and currently RDi. This guide assumes you will be working with the most recent version (as of this writing) of RSE as packaged in Rational Developer for i, RPG and COBOL Tools (aka RDi). RSE does come as part of some other packages from Rational, but RDi is the most common way of getting the toolset. There is a trial version of RDi available for download, so even if you're not yet ready to commit actual \$s for a license, download the trial version. It is completely functional and once you become as addicted to this great new way to write code as we are, you simply need to purchase a license and get an Activation Kit from IBM/Rational to allow it to continue to work after the trial period ends. If you download version 9.6.0.4 or later, your trial period will be 120 days. Earlier releases had a 60 day trail period. More on downloading and installing the trial follows.

If you have WDSC V7 instead of RDi, most of what we describe in this guide is still applicable to you, with the exception of the installation portion. However, WDSC is no longer supported by IBM/Rational and the language support in the editor in WDSC is frozen at V5R4.

### Why Move to RSE/RDi?

Before we get into the “how” part of the guide, we thought we'd spend just a little time on “why.” We have been RSE users - and fans - for years and like most RSE users we find ourselves wondering how we ever managed to write code without it. Here's a short list of some of our favorite features, just to give you an idea of the kinds of things we think you're going to love in RSE.

- More lines of code on the screen (typically 2 to 3 times what SEU shows)
- Filtered source code - by change date, to remove comments, show control flow, etc.
- Unlimited levels of Undo & Redo - even after a save and compile
- Tab keys work without prompting (and can be customized)
- Multiple source members open for edit at once
- An Outline view cross referencing all fields (internal and external), subroutines, etc.
- Compile error feedback without closing the source and without spooled file listings

We could (and often do) go on and on with our list of favorite features, but hopefully that gives you an idea of what you have to look forward to. So let's get started!

## Downloading and Installing RDi

RDi is installed on your workstation and works across the network to attach to your IBM i host (or iSeries or System i or AS/400 - whatever you call that system where your source members are and where your code compiles. We'll call it "IBM i" in this Guide.)

RDi currently runs on Windows and on Mac. We run ours natively on our Macs. It is also available for a few versions of Linux but most people use it with Windows.

You can download RDi from the IBM web site (<https://www.ibm.com/developerworks/downloads/r/rdi/>). You will need an IBM ID. If you don't have one, you can create one as part of the download process.

There are 2 ways to do the installation. You can either install directly from the web (IBM's site may call this "Web install using Installation Manager") or you can download 2 (or more) very large zip files for the download. The recommended approach - especially if you are downloading for just 1 or a few computers - is to use the web installation option. Using the web installation method also ensures you will get the latest fix pack level with its 120-day trial period. If you want to use the full download method of installation instead of the web install, make sure you follow the recommended link to the "technote" from IBM's web page.

After pressing the "Download trial" button you will be prompted for whether you want the web install for Windows, the web install for Mac or Linux or the full download installation method which includes support for all 3 platforms. If you are not already signed into the IBM web site, you will be prompted to sign in. Eventually you will see a "Fix Central" page where you can download the web install zip file. Note that although the file you're downloading says it is 9.6.0.0, it will still download the latest version (as of this writing the latest is 9.6.0.4.)

Following the initial installation, as with most other workstation-based products, it is highly recommended that you immediately apply any available updates. Updates are applied using the IBM Installation Manager. Run the Installation Manager, choose Update and follow the instructions for the recommended updates (if any).

## Starting RDi

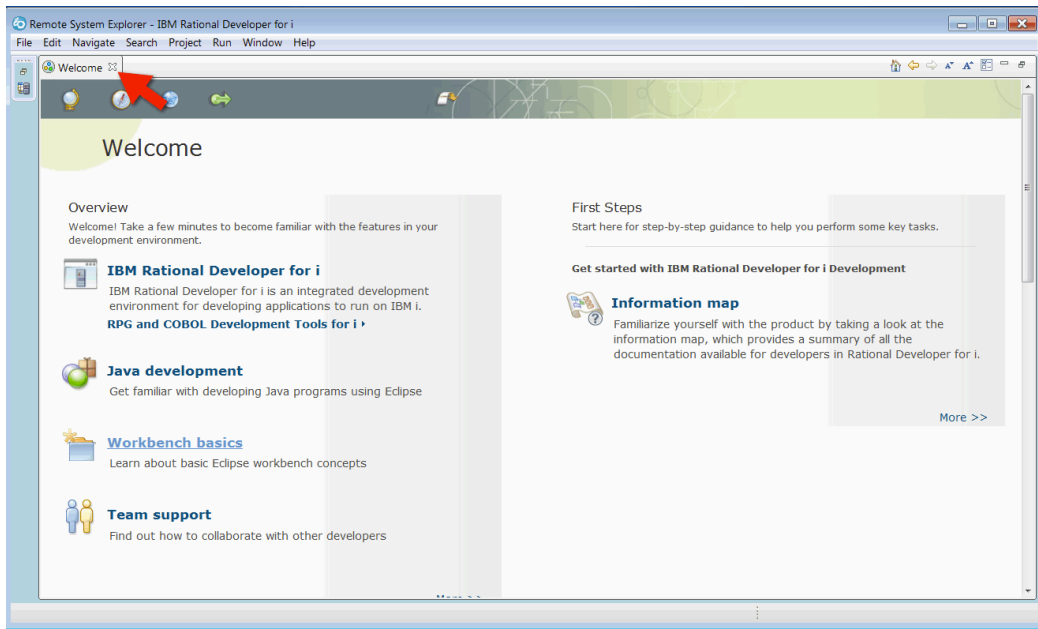
Finally, we get to start RSE. From Start Programs or your favorite form of shortcut, start IBM Rational Developer for i (probably under a menu called IBM Software Delivery Platform.)

Assuming this is the first time you've started it after installation, the first dialog you should see is a small window called the Workspace Launcher. The workspace is a set of directories and files used by the workbench to store information about your RDi workbench environment. It is a critical part of using RDi and it will be accessed very frequently. For performance reasons it is important to make sure the workspace is on a local drive on your

own workstation - not a shared network folder. We also recommend putting your workspace in a simply named folder on your drive (e.g., c:\RDiWorkspace). This just makes it easier to locate in case you need to at some point in the future. If the folder doesn't already exist, it will be created for you.

After specifying the folder location you want, you may want to check the box "Use this as the default ..." to avoid seeing this dialog again. If you ever want to change your workspace or create a new one, you can do so in the RSE workbench via the File menu and the Switch Workspace option.

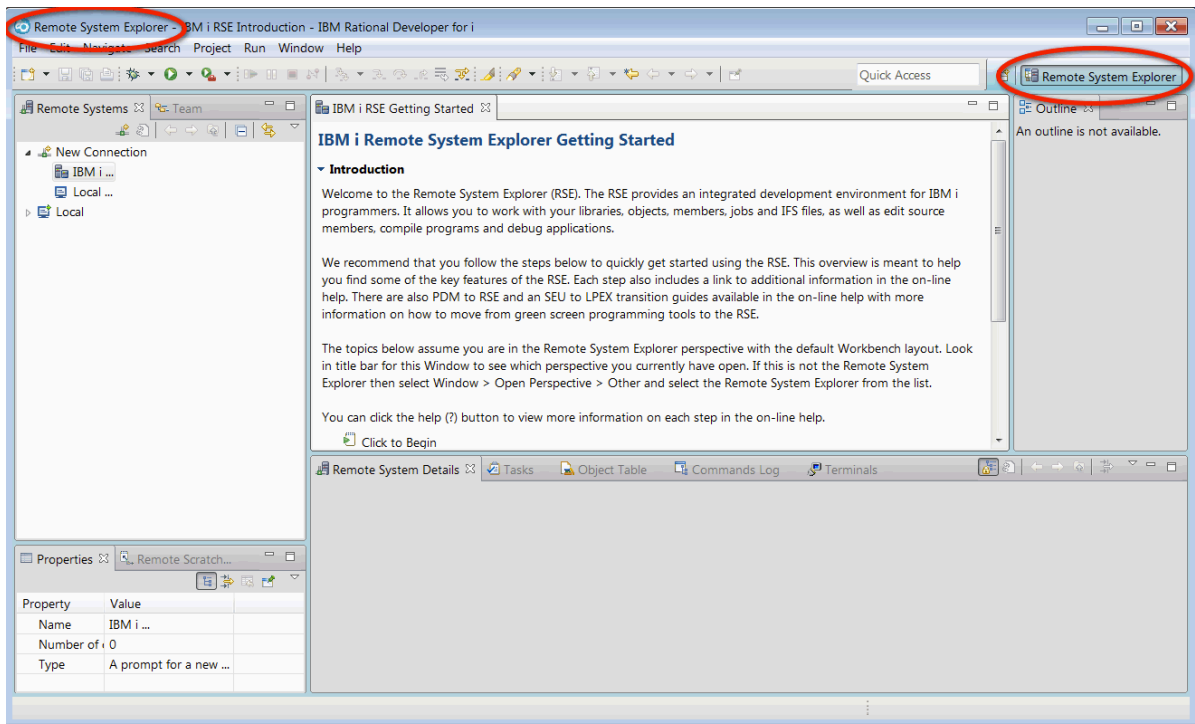
The workbench will start after you have specified your workspace location. The first time it comes up, you will most likely see a Welcome splash screen appear that looks like the one below. If you see a message box about features with trial licenses, use the Ignore Button for now. You can apply a license later (if/when you purchase one) from the workbench by pressing the "Manage Licenses" button. If at some point you see a dialog about initializing Help, take the "Run in Background" option.



From the Welcome screen, you can access introductory information and tutorials about the RDi workbench (click on the More >> link on the left side). You may want to come back to these at some point in the future. In the interest of getting a "quick start", we suggest you close this screen by clicking on the small "X" on the "Welcome" tab near the top left of the screen highlighted in the figure above. If you want to look at this information later, simply take the Welcome option under the Help pull down menu in the workbench.

You should now be in the workbench. By default, the workbench starts in the RSE (Remote System Explorer) perspective immediately after installation and should look something like the picture below. Note that the window bar title shows "Remote System Explorer..." and it also appears near the upper right corner of the workbench, as highlighted below (although as

of version 9.6, the icon in the upper right may not have the text - just the icon itself. You may see the “IBM i Remote System Explorer Getting Started” document in the editor window in the middle. It will probably appear once you begin to create your connection in the next few steps.



If you are using a workstation that has opened the workbench before and you see something besides “Remote System Explorer ...” in the areas highlighted above, you can open the Remote Systems perspective by the following steps:

1. Select the *Open Perspective* option from the *Window* pull down menu
2. Select *Remote System Explorer*

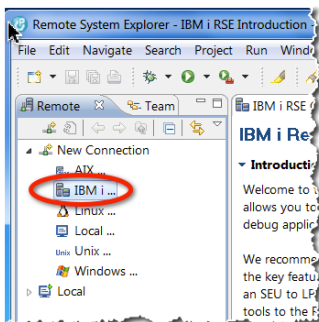
A perspective is a specific arrangement of tools in the workbench. Depending on the role(s) the workbench user has, he/she may use a number of different perspectives. For example a web developer will use the **Web Perspective**, a Java developer will use the **Java Perspective**, and a developer writing IBM i host-based applications (in RPG, CL and/or COBOL) will use the **Remote System Explorer Perspective** (but will likely eventually also need the **Debug Perspective**.) Of course, quite often, one developer has many roles and will use multiple perspectives. We’re only concentrating on Remote System Explorer here.

Make sure your workbench window is big enough to be useful when things begin appearing in the window panes you see here. Use the maximize button on the top right of the workbench window or stretch the corner of the window to give yourself plenty of room to work.

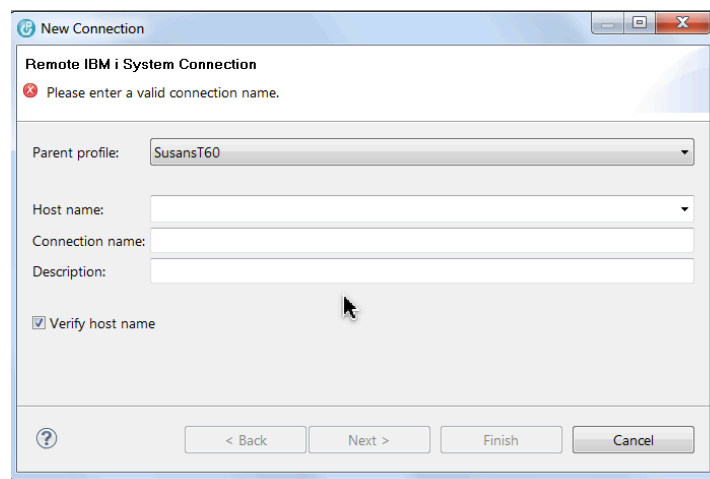
A few basics about navigating the workbench are in order. You have noticed by now the different window “panes” within the workbench. Each pane is called a View and contains a different development tool – sometimes more than one tool. Each view can be used in its small pane form as it appears now. In addition, each can be easily blown to full screen size. Do this by double clicking on the bar (where the file tabs are) at the top of the view’s pane. Try this by double clicking on the tab that says “Remote Systems” (being careful to avoid the X in the tab.) You should see the Remote Systems view occupying the full space of the workbench. Double clicking in the same place again shrinks the view back to its original size and position. Try your hand at maximizing some of the other views. The editor view in the middle is the view that you will most commonly want to use in full screen mode so that you can see more code on the screen. You may also stretch any of the borders between the views with your mouse to change their relative sizes and shapes within the workbench.

## Creating a Connection to Your Host System

Now, for the moment of truth! Let’s connect to your host IBM i system. To do this, we will create a Remote Systems IBM i connection. In the Remote Systems view, locate the **New Connection** node as shown here. If it is not expanded, expand it (by clicking on the + sign or arrow beside it.) Don’t be concerned if you don’t see all the connection types shown in this image. All you really need (for now, anyway) is IBM i.



Locate the **IBM i ...** node and double click on it. The new connection wizard will appear:

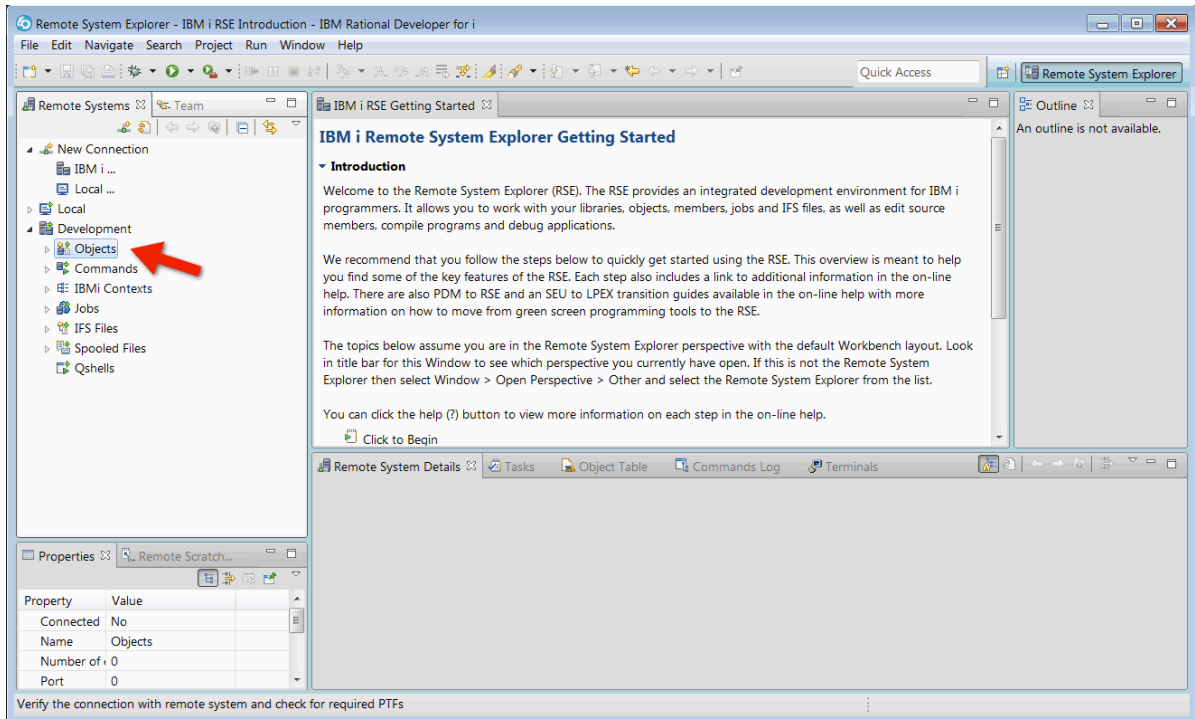


Note that the “Parent Profile” has nothing to do with your host user profile. It defaults to the name of your workstation and, at least for now, it’s best to leave that at the default.

The cursor in the **Remote IBM i System Connection** dialog should be positioned in the **Host Name** entry field and it may contain LOCALHOST at the moment. Enter the host name (or the TCP/IP address) for your IBM i as it is known on the network (replacing LOCALHOST.) The **Connection name** is the name that you will see in the Remote Systems list in the workbench. If you don’t fill in a name here, it will be automatically filled with the host

name. Fill in a name here that you will easily recognize for this system. Click the Finish button to complete the connection.

You may create as many connections as you like. A connection is exactly what its name implies. It represents a connection to your host system and when activated results in a server job (sometimes more than one) on the host system. Since you can supply any name you like, you may want to have connections to multiple IBM i systems (or LPARs) or multiple connections to the same system. Each connection may have a default user profile and other properties, such as a default library list.



After creating your connection, your Remote Systems view should look something like the one above (we have used “Development” as our Connection name.) Before you attempt to use the new connection, you should first verify it. This will alert you to any PTFs that may be needed on the host system.

Under your connection name you should see **Objects** (it is selected in the image above). Right click on **Objects** and select **Verify Connection** from the action menu. This will connect to your host system and check for missing products or PTFs. Since it is connecting to your IBM i, you will be asked to sign on with your normal host User ID and password. If this is the user ID you plan to use for this connection normally, you will probably want to check the boxes to save the information. If verify connection fails to connect to your system, you probably have an error in the Host Name. To change it, right click on the connection name and select Properties from the action menu. If the "Verify Connection" connects and lists PTFs that are missing, you should have those PTFs installed, at least those that are related to the Remote System Explorer. You may find failures reported on callback daemons. Some of these are for processes that have recently been changed, but the checks for them have not yet been removed from the verify connection process. As long as you seem to have the PTFs

necessary – particularly those for Remote System Explorer and the Integrated Debugger – you can proceed with using your connection.

## Let's Get to Work!

You should now be ready to use your RSE connection. Note the “subsystems” listed under your connection name: **Objects**, **Commands**, **Jobs**, **IFS files**, **Spoiled Files** and **Qshells**. (Note that the term “subsystem” in RSE has nothing to do with IBM i host subsystems.) In this guide, we will concentrate only on **Objects**, but you may want to explore some of the others on your own.

Expand the **Objects** subsystem (if it is not already expanded) by clicking on the arrow or the + sign. The first three entries (Work with libraries, etc.) are reminiscent of PDM. However, they don't work quite the same as PDM. They are like a prompted “Work with ...” command in PDM but each filter can be named and saved for future use. You will want to create your own filters eventually, but for now, in the interest of getting you off to a “quick start,” we're going to skip over these.

Let's begin by using one of the default filters supplied with RSE: the **Library list** filter. By the way – a word of warning – **avoid** expanding the **User libraries** supplied filter. On most systems, the number of user libraries is so large that expanding this filter leads to excessive time spent twiddling thumbs waiting for the list to appear! And when it finally does appear, it's too long to be useful.

Expand the **Library list** filter in your connection. If you didn't save your user ID and password earlier when you verified your connection, you will be asked for them again here (and many more times later, so now would be a good time to save them). You will probably notice a delay as the workbench goes to the IBM i to create a server job for your requests and retrieves the library list for that job. You should eventually see a list of libraries. This is the default library list for your user profile.

If this list doesn't look like the library list in your interactive jobs, it's probably because you have an initial program that runs after you sign on to change your library list. You can customize your initial library list later using the **Properties** of your **Objects** subsystem. We'll show you how to do that in a later section of the guide.

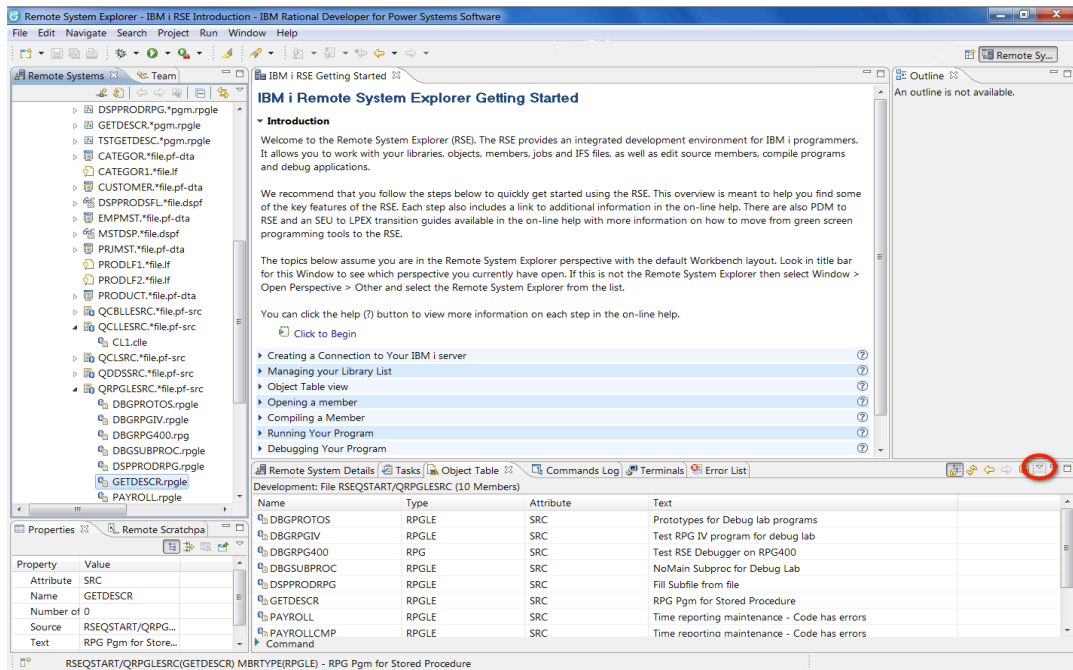
Note that you can expand a library to see its objects. Pick a library (preferably a relatively small one and/or your personal library). If the library you want to work with does not appear in the **Library list**, you may add it (both to the filter *and* to the actual library list of the server job on the host) by right clicking on the **Library list** entry and selecting “Add Library List Entry ...” or “Change Current Library...” The wizard will step you through these actions. Note that changing your library list this way only affects the server job you are connected to at the moment. If you were to disconnect from the host and reconnect later, you will need to adjust your library list again, much as you would do if you signed off from an interactive



session. If you want the library list to be maintained permanently, use the properties dialog mentioned earlier.

Many objects can be expanded as well. In our case, look for a source physical file (QRPGLESRC, for example) to expand. Note that the objects appear in the same sequence they would in PDM – i.e., \*Pgms come before \*Files and are in alphabetical sequence within object type. If the list seems too long to work with productively, you now know why you have the flexibility to create your own filtered lists, as we discussed earlier. Remember that you can always make the Remote Systems list full screen by double clicking on the tab at the top of the view.

Find and expand a source physical file to show the list of members. You'll notice that you have the source type and member name listed. If you are a big user of other information from PDM, such as the member text, try this: Right click on the source file name (e.g., QRPGLESRC) and select “Show in table”. You will see a list of members appearing in a separate view below the editor - the **Object Table** View. If a command entry area obstructs part of the list at the bottom, you can close it by clicking on the blue arrow just above the word “Command”.



Note: The image above shows the Object Table as it appears by default at version 9.5.2 and earlier. In version 9.6 and later, you will see an option column on the left side and a "Filter/Subset" dialog on the right side.

You will most likely want to double click on the **Object Table** view tab to take it to full screen. Now you have a more “PDM-like” view of the members in the file, with a couple of twists. For example, click on a column heading in the table and it will sort the members in that sequence. Click on the small down arrow (the view menu arrow) at the top right corner

of this view (its position is highlighted in the picture above) and select **Show Columns**, then select **All** from the resulting menu. Voila! You now have Date Created and Date Last Modified in your member list as well. You may need to stretch the column headings to see all the information. **Object Table** view works for libraries and objects as well as members. You can choose whether to interact with your objects/members from the table view or from the **Remote System** view. Right clicking on a member in either list shows the same options available to you and double clicking will open the member for edit.

Using either the **Object Table** view or the **Remote Systems** view, open a source member in the editor. The easiest way to do this is to double click on the member name. Alternatively, you can right click and select “Open with”, then “Remote Systems LPEX Editor”. Note that if you don’t like drilling down to find a member to open it and you know the precise name of the member you need to open, you can always right click on **Objects** (in the **Remote Systems** view) and select **Open Member...**

If it’s a very large member and/or if your connection to the host is not speedy, it could take a while for the source to appear in the editor view in the middle of your workbench. While it does take longer to see the source in this editor compared to SEU, once it is open here, it is now local. This means you can go from the top to the bottom of the source as fast as you can drag the scroll bar -faster still if you use the keyboard shortcuts in the edit view Ctl-Home (top) and Ctl-End (bottom) – and anywhere in between. We’ll look at a few of the many other navigational features of the editor later. Also, you don’t need to close a member to open another one, nor do you need to (or want to) close it to save it or to compile it.

Before we go on to look at the editor, go back to either **Remote Systems** view or **Object Table** view and choose 2 or 3 other source members to open –from the same or a different source file. Do NOT close the first source member you opened. One of the greatest features of the workstation editors is that you can have multiple members open at once. Open an RPGLE source member if you have one, as we’ll be using that language in our editor example a little later. Note that when a member is open for edit in this editor, it is locked on the host to avoid others changing it at the same time - just as SEU locks open members.

Notice that when you have multiple members open, they each have a tab at the top of the editor view. You can simply click on the tab with your mouse to move from one to another. If you prefer to leave your hands on the keyboard, you can use a keyboard shortcut – either Ctl-F6 or Ctl-E.

## **The Editor**

Let’s take a look at just a few of the productivity features in the editor. We’ll be using the RPGLE (RPG IV) language to point out some RPG-specific features, so if you have an RPG IV source open, you can follow along with your own source.

Double click on the tab at the top of the editor screen to make the editor view full screen. Since we’re about to make some changes to this code, you may want to make a copy of this



choosing a specific date range, for example. This can be handy if you suspect a problem was introduced in some recent changes to a program. You can also filter to see only the subroutines or procedures or only the operations that control logic flow. Whenever looking at filtered view, use right click > **Show all** or **Ctl-W** to get back to the full source.

Notice the format line at the top of the edit screen showing, in our example above, a calc spec. The current cursor position is highlighted with a reverse image block to show where in the spec my cursor is located. If you move your cursor to another spec type, such as a D spec, you will see the format line changes to the spec type where your cursor is. Even better, you'll notice that you can use your tab key directly in the editor to get from one part of the spec to another without prompting. We find we rarely, if ever, use prompting in the RSE editor even though we often used it back when we used SEU.

Using either your mouse or the keyboard, select (highlight) a few lines of code in a block (**Tip: Alt-L** is a shortcut to select one or a block of lines). Then use the clipboard to cut those lines from their current position and paste them elsewhere in the source. You can choose **Cut** and **Paste** from the **Edit** menu, but, of course, you can also use the standard keyboard shortcuts **Ctl-X** and **Ctl-V**. The SEU-style line commands work here as well (C, CC, D, A, B, etc.) but being equally comfortable with both ways of editing makes you more productive.

If by chance you didn't quite line up the paste point exactly, you may have received some errors from the syntax checker in lovely bright pink. Don't worry – either just ignore the messages until you can move things around or, if you prefer, you can delete all inserted error messages from the view by using the keyboard shortcut **Ctl-F5** (or use the **Remove Messages** option from the **Source** pull down menu.)

Don't worry too much about the fact that you've made some changes to this source member and made a mess of it, because you also have available one of our most favorite features of all – the **Undo** option. Use the **Undo** option from the **Edit** menu or (more likely if you use it as much as we do) use the **Ctl-Z** shortcut. Undo all your changes and the editor will notify you when the source is back in its original state via the message line at the bottom of the editor view. By the way, there is also a **Redo** option (**Ctl-Y**) in case you undo too much. It sure beats F3 and “no save” in SEU, doesn't it?

The source code we're showing here uses fixed-format RPG. We prefer working in free format RPG in real life. If you want to convert to free format, there's an option under the **Source** menu, **Convert All To Free-Form**. There are many things that will not convert with this very basic option, but if your code is written in a fairly modern style, it will likely do most of the work for you - at least for the logic portion of the program. Feel free to give it a try, because you can always undo (**Ctl-Z**) that as well! Note that this built-in basic conversion option only converts C specs and it is a very simplistic converter - any operations it cannot convert it will leave as fixed format code.

Note that there are plug-ins you can install for this RPG editor that will do a far better job of converting existing code to free format - more complete, including H, F, D and P specs and

far better at converting nearly all the logic. Those plug-ins have a charge for them but if you want to convert lots of code to free form you should check them out - the ones we're aware of are from Help Systems (formerly Linoma Software) RPG ToolBox and ARCAD Software's ARCAD-Transformer RPG.

We could go on for hours showing you all our favorite features of the RSE editor, but then we'd have to rename this to the Not-so-Quick Start Guide. But there are few more items we really want you to see. Position your cursor on a calc spec line and press F1. It may take a while to react, especially the first time you use it, but eventually a second window will appear with Help in it. Notice the help text on the page. It should be a page of the RPG reference manual - either a list of operation codes or the operation code on the statement where you pressed F1. That may not be what you wanted information about, but you can navigate through the *Contents* on the left side of the Help window, allowing you to be at the appropriate page of the reference manual in a matter of seconds. It tries to be intelligent about where it positions you in the manual. For example, if you position your cursor in the keyword area of a D spec, it will position the manual to the D spec keywords area of the manual.

Being able to get to the page of the reference manual that you need almost instantly means you're more likely to use new, more powerful language features, which in turn increases your productivity and makes your code better at the same time.

**Tip:** Since the help window takes a while to open, get into the habit of leaving it open when you go back to the editor screen. You can also get directly to some of the most useful parts of the manual from the Source pull down menu in the editor, option ILE RPG Help. Of course, there is also access to the DDS, COBOL, CL and SQL reference manuals if you're working in those languages - check out the *Contents* column to find them.

Back in the editor, make sure your source is back at its original state. **Tip:** A quick way to tell if there are any unsaved changes in your open source member is to look at the name of the member in the tab at the top of the editor window. If there is an asterisk (\*) just before the member name, there are unsaved changes. You could also try Undo to see if there are any changes.

### **Finding Compile Time Errors**

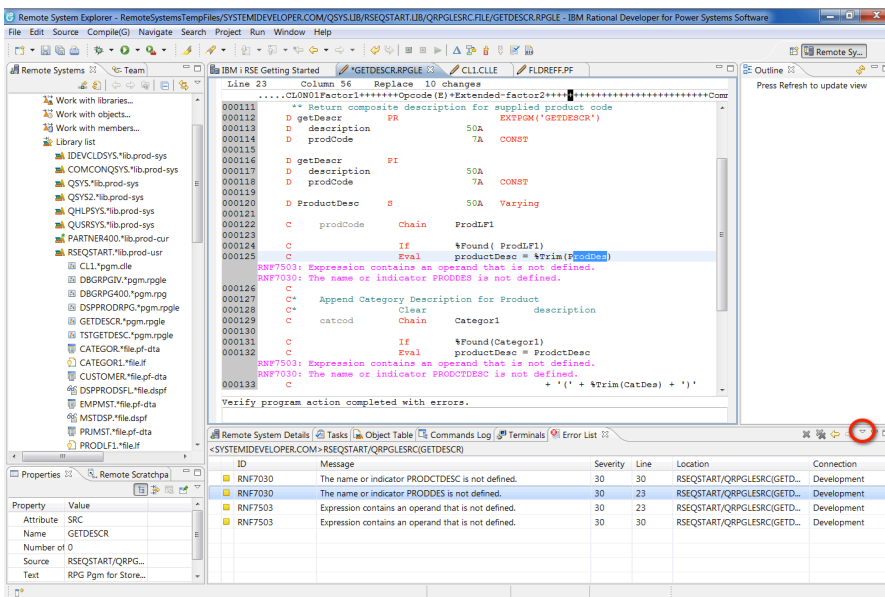
We're assuming that the source member in its original state probably can be successfully compiled, i.e., it has no errors in it. Note that you may need to add 1 or more libraries to your library list for the compiler to find all the files and other objects it may need to do a successful compile. To demonstrate the error feedback built into the editor, you should make 1 or 2 intentional errors in the code. Choose errors that are not simply syntax errors, but ones that only an attempted compile would find, such as misspelling the name of a field or subroutine.

Before compiling, if your editor is in full screen mode, double click at the top to get back to the multi-window workbench view. This will allow you to more easily see the compile errors that will appear below the editor window.

Once you know you have at least one “compile type” error in the code, use the appropriate type of compile command (e.g., CRTBNDRPG or CRTRPGMOD for RPGLE) from the Compile menu. If the compile command you want is greyed out in the menu, click back into the editor view and try again. The tab of the source member you want to compile must have focus.

After the compile has finished, you should see some messages appear in the **Error List**, a view that overlays the **Object Table** view we used earlier. You will probably see some messages that are informational (severity 0) only.

**Tip:** Since we rarely need to see all those “Name or indicator is not referenced” messages, we like to filter them out of the list. To do this, click on the tab for **Error List**. Click on the little down-arrow menu icon near the top right of the **Error List** panel (it says **View Menu** if you hover your mouse over it and it is highlighted in the figure below). From the menu, select **Show Severity** and click "off" the **Information** messages. Now you are left with only “real” errors in your list. This setting will remain in place for all future verify actions and/or compiles until you change it.



Double click on one of the errors and it will position your cursor in the editor to the line of code in error. You’re now ready to fix that line of code and go on to double click on the next error in the **Error List**. Isn’t that a little more productive than closing the member in SEU, answering Y to the save request, putting a 14 next to the member name, checking the job end message to see you have an error, selecting and opening the compile listing spool file from a list, navigating to almost the bottom of the listing to find the error list, paging up further to determine which line of code was affected, reopening the source member in SEU, and

manually navigating to the line number in error? You'll probably find, as we have, that you very rarely look at compile listings as an RSE user. We don't miss them at all.

After fixing the errors in the list, you can re-compile from the Compile menu as before. Note that you have the option to either prompt your compile commands or not from that menu. You may also compile without prompt using the shortcut Ctl-Shift-C.

## Outline View

**Note:** Our Guide is currently written for RDi Version 9.6. If you have an earlier version installed, your Outline view behavior will be a bit different.

The **Outline** view is one of our very favorite RSE features. It's hard to remember how we ever coded in RPG without it! The **Outline** view (by default) appears to the right of the editor window. Note that the Outline is most useful for RPGLE members, somewhat less so for DDS members. It can be used with some other member types such as COBOL and CL, but currently with far fewer useful features.

Your Outline view will automatically be populated when you open a member that supports it. If your Outline were to be empty except for the words "Press refresh to update view" then you probably have a version earlier than V8.5. In that case, click on the Refresh icon, which looks like two yellow arrows forming a circle near the top right edge of the view to populate your **Outline**. Note that if you are at that earlier release, you will also need to press that refresh icon periodically after making changes to an RPG program to keep it up to date. On the current release, all changes made to the RPG source automatically update the Outline.

You should now see expandable lists of files, fields, procedures and subroutines, among other things. If there many files, /Copy members or other external references used in this member, initial population of the **Outline** could take quite a while. This is because RDi is retrieving all the external descriptions and /Copy members. Fortunately, the results will be cached for future use when you're editing this source member so future refresh actions will go much faster by using the cached information.

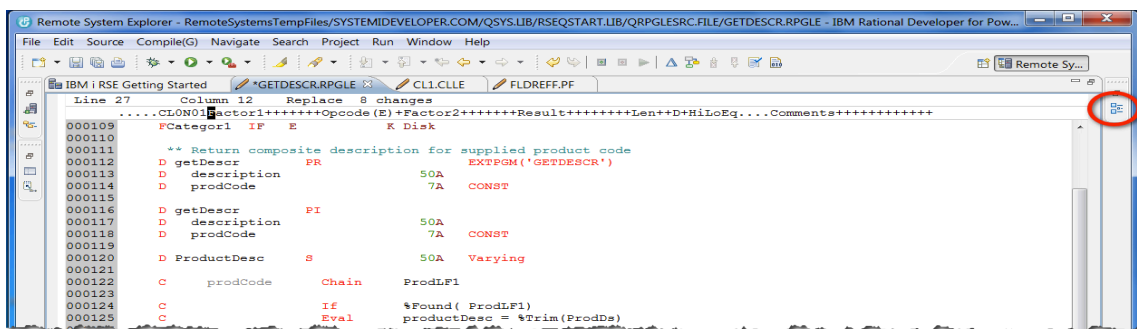
Expand some of the lists to see the gems hidden here. The Fields list, for example, contains a list of every field in the program along with its data type and size. You even get a cross reference to the lines of code where each field is referenced. Statement numbers with (M) at the end are lines where the value of the field is modified. (See an example of the **Outline** view in action in the figure below.) The cross reference list is connected to the editor, so if you click on a statement number, your editor window will be positioned to that line. Clicking on the field name itself positions you to the line where it is defined. Now, that's handy!

We'll leave you to explore the Outline view in more detail on your own. We could probably write a whole guide just on all the great uses for the Outline! Here are just a few tips for you in your explorations.

**Tips:** By expanding a file (PF, LF, DSPF or PRTF) and then its record format(s), you can get all the external definitions, including keys, without the need for using DSPFD or DSPFFD or opening the DDS source. Click on subroutine or subprocedure names to navigate quickly to those parts of the code. Hopefully you use prototypes for all your calls in RPGLE. If so, then expanding the prototypes gives you the parameter lists!

## Fast Views

You hopefully remember (and have been practicing) double clicking on the top of a view to put it into full screen mode (and back again). This is particularly helpful for your Editor view so that you can see more code while editing. However, you can't see your other views when using a full screen editor view, which means that great Outline view we just learned about is not readily available. Never fear – that's what Fast Views are all about.



When you go into full screen with the Editor (or any other view), notice the small icons around the margins. In particular, notice the **Outline** icon (highlighted in the image above). Click on that icon to pop out the Outline view over your editor. Very handy! There is no need to leave full screen editing to use your Outline - nor, for that matter to use any other view, such as Remote Systems to open another source member.

**Tip:** It's very important that when you're finished using the Fast View, you **do not** click on the x in tab to close it. Instead, simply click back into the Editor to continue editing your code. The **Outline** (or whatever Fast View you just opened) will automatically close. If you do forget someday and accidentally close the **Outline** (or any other view), use the **Show View** option from the **Window** menu. Then choose the name of the view, or **Other** and find the view you want there (such as Outline).

## Customization

We have already covered a few customizations that we like to use in earlier parts of this guide. But here are a few more that we like to do for new installations.

We find the editor's default behavior of repeating the spec type on a new line every time we press Enter rather annoying. Of course, now that RPG is (almost) completely free, for new programs this isn't much of an issue. But we suspect we'll all be maintaining fixed form specs for a while yet. We prefer to have it just put in a new blank line. Making that change is a simple preference setting: From the **Window** menu, choose **Preferences** (probably the last



option). In the resulting dialog, your cursor will be in a box with the text “type filter text” highlighted. This is a search box to help you navigate through the hundreds of preference options. Key **RPG** into the search box, then select the **ILE RPG** option under **IBM i Parsers** (not the one under Program Verifiers). Here you may, if you want, de-select “Repeat previous specification type.”

On the same RPG preferences page referenced above, notice that you may customize the tab stops for any spec type. This can be quite useful, particularly in fixed format RPGLE code since we never use conditional or level indicators on C specs and, for that matter, even Factor 1 isn't used all that much. You could set up your tabs to go directly from the C (when you're not using Free format) to the operation code with a single tab key. The tab stops are marked with the ^ symbol - simply move or remove any stops you don't want.

**Tip:** Get accustomed to using the tab key in the editor - much more productive than prompting. Follow your current cursor position on the format line at the top of the editor screen, which changes as your cursor moves from one spec type to another.

We mentioned earlier that you could change your default library list by editing the properties of your connection. If you want/need to do that, let's give that a try now. In the **Remote System** view, right click on **Objects** immediately below your connection name. Choose **Properties** and in the resulting dialog choose **Initial Library List** in the left pane. Here you have several options for controlling your default initial library list.

Near the top, you can enter 1 or more **Library** names that you want to be added to the library list that is the default for your user profile each time RSE starts. You also have the option of specifying a **Current library**. And you can specify an **Initial Command**. Many developers have an initial program specified in their user profile which is called when an emulation session is started. A similar initial program can be used here - just be sure to put CALL before the program name since this is a command. You should probably take a look at that initial program and make sure it doesn't do anything that requires an emulation session (such as STRPDM) since, of course, in RSE those operations will not work. You may need to make an edited version of your initial program for RSE purposes.

## Keyboard Shortcuts

We find that keyboard shortcuts, especially in the editor, can be more productive than using the mouse. You can get an incredibly long list of keyboard shortcuts for the SEU profile (which is the default profile for most IBM i languages) by keying “? SEU” into the command line box at the bottom of the editor view (press the Esc key while in the editor view to get there quickly). We've already mentioned several shortcuts in this guide. Some of those are summarized below (there are a few more of our favorites here as well).

We recommend that you download (or request a hard copy) of a larger list of “System i Developer's Favorite Keyboard Shortcuts for RSE” here: [www.systemideveloper.com/downloads.html](http://www.systemideveloper.com/downloads.html).

### **Windows shortcuts:**

Ctl-Z (undo) and Ctl-Y (redo)

Ctl-C (copy), Ctl-X (cut), and Ctl-V (paste)

Shift key plus the right/left/up/down arrow (select code)

Ctl-F (find/replace)

Ctl-Home (top of source) and Ctl-End (bottom of source)

Ctl-S (save, back to where the file was opened from - in RSE, typically the IBM i host)

### **RSE Editor specific shortcuts:**

Ctl-F5 (Clear error messages from editor window)

Ctl-Backspace (Delete current line, such as one put there accidentally by the Enter key!)

Ctl-W (Show all lines, e.g., when looking at a filtered view in the Editor)

Ctl-Shift-V (Verify source)

Ctl-Shift-C (Compile source)

Ctl-F6 (Next Editor tab/ next open member)

Alternatively Ctl-E (List of editor tabs/open members)

Ctl-M in Editor is Match beginning/end of block to other end

Like on If ... EndIf, Dow ... EndDo, etc.

Alt-S (Split the current line at the cursor position – useful for long expressions)

Alt-J (Join the next line to the end of the current one)

### **Closing Down RDi at End of the Day**

What should you do with RDi when you're finished using it for the day? Here's what we recommend. By the way, we recommend that once you have RDi open, leave it open for the rest of the day. It does take a while to start up, so why not leave it there, ready to go for a quick look at some code whenever the need occurs? Otherwise, you may find yourself tempted to jump into SEU (shudder!) if you're in a hurry. Then, at the end of the day:

1. Close any source members you have open in the *Editor*. To close a source member, click on the x in the tab next to the member name at the top of the editor. If there are unsaved changes at the time, the editor will ask you if you want to save them. You can also save from the File menu or using the Ctl-S shortcut. Closing the members not only reminds you to save the changes back to the host, but it releases the lock you've held on each source member to keep others from editing it at the same time.
2. In the *Remote System* view, right click on the connection name(s) you have using and choose *Disconnect*. This will end the job on IBM i that has been servicing your requests for lists of things, source members, etc.
3. If you shut your workstation down each day, you can close RSE using the X in the top right corner. If you don't shut down your workstation, you may want to leave RSE loaded - ready for tomorrow's programming challenges! Just lock your workstation as you would normally.

## What Next?

To borrow a phrase from Nike - Just do it! Anything new feels strange for a while. One way to become more comfortable with these new tools is to simply use them. If you don't feel that you can quit SEU cold turkey, try using RSE for 2 hours each day. Chances are, you won't have trouble convincing yourself to extend that time to 4 hours a day after a week. Before long, you may join us in wondering how you ever wrote code without these tools!

We've intentionally included only the basics in this Quick Start Guide to get you using it as quickly as possible. You may want to read it again after using RDi for a few days because you may have forgotten some of our tips and hints.

There is much more to learn about RSE. We have only scratched the surface. After using it for many years, we still learn new things about it from time to time. There are additional tips for getting started in the "IBM i RSE Getting Started" document in the tool itself. We (and others) have written many articles about various features of RSE and RDi that are readily available via the internet. We also teach classes to help people move from SEU to RSE and RDi - both online and in person with many hands-on practice exercises. Contact information is below if you would like some customized training for your company.

If you get stuck, ask for help - from us (see below) or from other RSE users. Some good places to find other RSE users include:

- The WDSCI-L mailing list on Midrange.com <http://lists.midrange.com/mailman/listinfo/wdsci-l> - Don't be concerned about the "old" WDSC name - there are lots of RDi users there.
- IBM DeveloperWorks - The RDi Hub Forum <https://www.ibm.com/developerworks/community/forums/html/forum?id=11111111-0000-0000-0000-000000002285>

Please contact us with questions or feedback and suggestions on the RSE Quick Start Guide. We intend to create an FAQ related to RSE and the Quick Start Guide on the System i Developer web site so everyone can learn from the questions we receive. Contact us via the web site: [www.systemideveloper.com/contact.html](http://www.systemideveloper.com/contact.html) (there's a link there to email Jon and Susan.)

We hope you will find RSE as productive and useful as we have. Good luck with your explorations.